POLITECNICO DI TORINO

Corso di Laurea Magistrale

in MECHATRONIC ENGINEERING

(INGEGNERIA MECCATRONICA)

Tesi di Laurea Magistrale

Subject extraction and keyword extraction from text

Relatore/i

prof. LAVAGNO LUCIANO(DET)

firma del relatore (dei relatori)

....

Candidato/i

SONG ZERUI

firma del candidato

SONG ZERUI
....

Co-relatore/i

DENIS PATTI

firma del co-relatore (dei co-relatori)

Denis Patti
....

A.A.2019-2021

Abstract

A topic model is a model of people discovering abstract topics in a series of documents. With the topic model, people can more easily understand the topic of the document. In this thesis, we used methods based on statistics and neural networks to analyze their themes in a series of documents and evaluate their results. Finally, we show that the method based on neural networks is significantly better than the method based on statistics.

1 Introduction

Language is the essential characteristic that distinguishes humans from other animals. Among all living things, only humans have language ability. A variety of human intelligences are closely related to language. Human logical thinking is in the form of language, and most of human knowledge is also recorded and passed down in the form of language. Therefore, it is also an important, even core part of artificial intelligence.

Communicating with computers in natural language is what people have long pursued. Because it has not only obvious practical significance, but also important theoretical significance: people can use the computer in the language they are most accustomed to, without spending a lot of time and energy to learn various computer languages that are not very natural and accustomed; People can also use it to further understand the human language ability and the mechanism of intelligence.

Natural language processing refers to the technology that uses the natural language used by humans to communicate with machines to communicate with each other. Through artificial processing of natural language, the computer can read and understand it. The research on natural language processing began with the exploration of machine translation by humans. Although natural language processing involves multi-dimensional operations such as speech, grammar, semantics, and pragmatics, in simple terms, the basic task of natural language processing is to segment the processed corpus based on the ontology dictionary, word frequency statistics, contextual semantic analysis, etc. The unit is the smallest part of speech and is rich in semantic lexical items.

Natural Language Processing (NLP) is a subject that uses computer technology to analyze, understand, and process natural language. It uses computers as a powerful tool for language research. With the support of computers, language information carries out quantitative research and provides language descriptions that can be used jointly by humans and computers. It includes two parts: Natural Language Understanding (NLU) and Natural Language Generation (NLG). It is a typical borderline interdisciplinary subject, involving language science, computer science, mathematics, cognition, logic, etc., focusing on the field of interaction between computers and human (natural) languages. People refer to the process of using computers to process natural language at different periods or when the focus is different. It is also called Natural Language Understanding (NLU), Human Language Technology (HLT), Computational Linguistics HI (Computational Linguistics) , Quantitative Linguistics, Mathematical Linguistics .

The realization of natural language communication between humans and computers means that the computer can not only understand the meaning of natural language texts, but also express given intentions and thoughts in natural language texts. The former is called natural language understanding, and the latter is called natural language generation. Therefore, natural language processing generally includes two parts: natural language understanding and natural language generation. Historically, there has been more research on natural language understanding, but less research on natural language generation.

The topic model is a kind of NLU task, and LDA is a classic algorithm in the topic model.

The topic model was developed in the last century. In 1998, Christos Papadimitriou, Prabhakar Raghavan, Hisao Tamaki and Santosh

Vempala described an early theme model [1]. Thomas Hofmann proposed pLSA in 1999 and described the difference between pLSA and LSA (Latent Semantic Analysis) [2]. That is, LSA is mainly based on singular value decomposition (SVD) while pLSA relies on hybrid decomposition. He subsequently conducted a series of empirical studies and discussed the application of pLSA in automatic document indexing. His empirical results show that the performance of pLSA has improved significantly over LSA.

In 2000, latent semantic analysis was proposed by Jerome Bellegarda for use in natural language processing [3]. In 2003, Andrew Y. Ng and others proposed that the aspect model used for pLSA has a serious overfitting problem [4]. They proposed the hidden Dirichlet distribution (LDA), which can be seen as a combination of Bayesian thinking. pLSA. Perhaps the most common topic model in use today.

In this thesis, we first compare the effects of LDA and LSA in the topic model. Later, in the future work part, we tried some methods based on knowledge distillation, and its effect was significantly better than the result of LDA.

The thesis is also a project for the web platform "TiRaccontoUnaStoria" (which in Italian means I tell you a story). "TiRaccontoUnaStoria" is an academic activity, having the goal to record, collect, share and publish real life stories, by anyone who wants to share a memory, with friends, parents, or all the world.

More in detail "TiRaccontoUnaStoria" is cloud-based web platform with speech-to-text capabilities, in order to record, collect, search and listen to stories.

Its main features are:

- automatic transcription of oral stories, collected and recorded using a Telegram chatbot;
- automatic keyword extraction from any story: these keywords are then used to search and listen to the stories starting near the point where the keyword appears;
- automatic generation of a word-cloud made of keywords, which offers a synthetic and extremely intuitive visualization of the content of a story.

"TiRaccontoUnaStoria" is still an evolving project, in fact one of the future targets is make the system able to classify the stories with the purpose of strengthening the correlation of the stories stored inside the web platform.

2 Topic Model

Topic Model is a statistical model used to discover abstract topics in a series of documents in the fields of machine learning and natural language processing. Intuitively speaking, if an thesis has a central idea, some specific words will appear more frequently. For example, if an thesis is about dogs, words such as "dog" and "bones" will appear more frequently. If an thesis is about cats, words like "cat" and "fish" will appear more frequently. Some words such as "this" and "he" will appear roughly the same frequency in the two thesiss. But the reality is that an thesis usually contains multiple topics, and the proportion of each topic is different. Therefore, if 10% of an thesis is related to cats and 90% is related to dogs, the number of occurrences of keywords related to dogs will probably be 9 times the number of occurrences of keywords related to cats. A topic model tries to use a mathematical framework to reflect this characteristic of the document. The topic model automatically analyzes each document, counts the words in the document, and determines which topics the current document contains based on the statistical information, and what the proportion of each topic is.

The three main topic models include LSA, pLSA and LDA.

2. 1 LSA

The Latent Semantic Analysis (LSA) model first gave such a "distributed hypothesis": the attributes of a word are described by its environment. This means that if two words are close in meaning, they will also appear in similar texts, which means they have similar contexts.

Simply put, LSA first constructed such a "word-document" matrix: each row of the matrix represents a word, each column of the matrix represents an thesis, and the value of the i-th row and the j-th column indicates that the i-th word is in the There are several occurrences in j paragraphs or the tf-idf value [3,4] of the word, etc. After the LSA model has constructed the word-document matrix, for the following possible reasons, we will use Singular Value Decomposition (SVD) to find a low-order approximation of the matrix:

The original word-document matrix is too large and consumes too much computing resources. In this case, the similar low-order matrix can be interpreted as an ‘‘approximation’’ of the original matrix.

The original word-document matrix often contains a lot of noise, which means that not all of the information in it is useful. In this case, the process of finding the approximated matrix can be regarded as ‘‘noise reduction’’ on the original matrix.

The original word-document matrix is too sparse compared to the ‘‘real’’ word-document matrix. The so-called real matrix refers to a matrix that takes all the words and documents that appear in the world into consideration, which is obviously impossible. We can only get an approximation of the real matrix by analyzing a part of the data. In this case, the approximate matrix can be seen as a kind of ‘‘reduced’’ version of the original matrix.

2.2 Probabilistic Latent Semantic Analysis (pLSA) model

The Probabilistic Latent Semantic Analysis (pLSA) model is actually proposed to overcome some of the shortcomings of the Latent Semantic Analysis (LSA) model. A fundamental problem with LSA is that although we can treat each column of Uk and Vk as a topic, since the value of each column can be regarded as a real value with almost no limit, we cannot explain these values further. What does it mean, and it is even more difficult to understand this model from a probabilistic point of view. And seeking an explanation in the sense of probability is one of the core ideas of Bayesian inference.

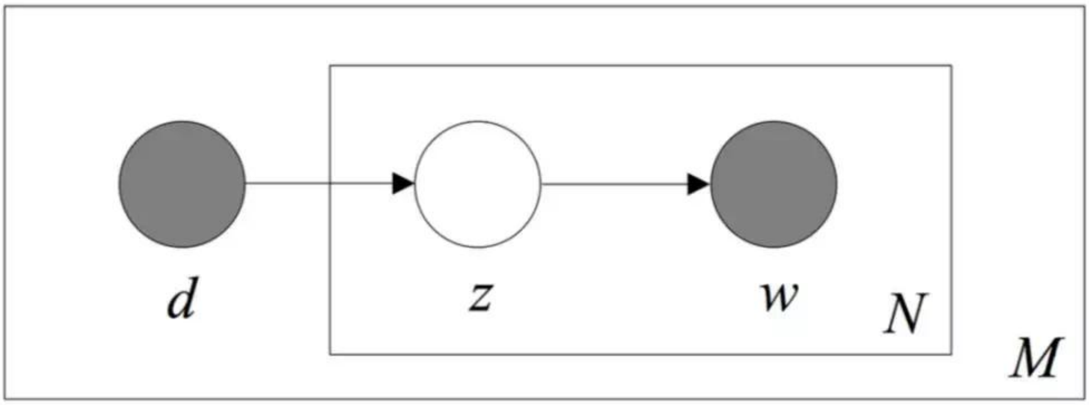
The pLSA model uses a generative model to give LSA a probabilistic interpretation. The model assumes that every document contains a series of possible potential topics, and every word in the document is not generated out of thin air, but is generated with a certain probability under the guidance of these potential topics [5].

In the pLSA model, a topic is actually a probability distribution on words. Each topic represents a probability distribution on a different word, and each document can be regarded as a probability distribution on the topic. Each document is generated through such a two-level probability distribution, which is also the core idea of the generative model proposed by pLSA.

pLSA models the joint distribution of d and w through the following formula:

$$P(w, d) = \sum_{z} P(z)P(d|z)P(w|z) = P(d) \sum_{z} P(z|d)P(w|z) \qquad (2\text{-}7)$$

The number of z in the model is a hyperparameter that needs to be given in advance. It should be noted that the above formula gives two expressions of P(w, d). In the previous formula, both d and w are generated by conditional probability under the premise of a given z , Their generation methods are similar, so they are ``symmetrical''; in the latter formula, d is first given, and then possible topics z are generated according to P(z|d), and then according to P(w |z) Generate possible words w. Since the generation of words and documents in this formula is not similar, it is "asymmetric".



The figure above shows the Plate Notation representation of the asymmetric form in the pLSA model. Where d represents a document, z represents a topic generated by the document, and w represents a word generated by the topic. In this model, d and w are observed variables, and z is an unknown variable (representing a potential topic).

It is easy to find that for a new document, we cannot know what P(d) it corresponds to. Therefore, although the pLSA model is a generative model on a given document, it cannot generate a new unknown document. . Another problem with this model is that as the number of documents increases, the parameter of P(z|d) will also increase linearly, which leads to the overfitting problem of the model no matter how much training data there is. These two points have become the two major flaws that limit the wider use of the pLSA model.

2.3 LDA

LDA is an unsupervised learning topic probability generation model. The input is the document collection and the number of topics, and the output is the topic presented in the form of probability distribution. It is often used for topic modeling, text classification, and opinions. Mining and other fields.

It assumes a premise: the document is equivalent to a bag-of-words, the words in the bag are independent and interchangeable, without grammatical structure and order.

The basic idea is: each document (Document) is composed of multiple topics (Topic), and each topic has multiple corresponding words (Word) to describe [6].

2.3.1 What is TOPIC?

Because LDA is a topic model, you must first clearly know how LDA views the topic. For a news report, we saw that it talked about yesterday's NBA basketball game, so we all know from the thigh that its theme is about sports. Why are our thighs so smart? At this time, the thigh will answer because there are keywords such as "Kobe", "Lakers" and so on. Well, we can define a topic as a collection of keywords. If these keywords appear in another thesis, we can directly judge that it belongs to a certain topic. But, dear reader, please think about it. What are the disadvantages of defining the theme in this way? According to this definition, we will easily give such conditions: once the name of a star appears in an thesis, then the topic of that thesis is sports. Maybe you immediately scold me for talking nonsense, and then retort that it is not necessarily. The thesis does have the name of the star, but it is all about the star's sex scandal, which has nothing to do with basketball. At this time, the theme is entertainment and it is almost the same. Therefore, a word can not be rigidly labeled as a subject. If there is a star's name in an thesis, we can only say that there is a high probability that it belongs to the subject of sports, but there is also a small probability that it belongs to the subject of entertainment. So there will be this phenomenon: the same word, under different theme backgrounds, its probability of appearing is different. And we can all be basically sure that a word cannot represent a theme, so what exactly is a theme? Students who can't stand their temper will say that since one word can't represent a theme, then I will use all the words to represent a theme, and then you can slowly understand it yourself. Yes, this is indeed a complete way. The theme is inherently contained in all words. This is indeed the safest way to do it, but you will find that this is equivalent to doing nothing. The old and cunning LDA thinks so too, but his cunning is that it uses very sleek means to express the subject in all vocabulary. How is it smooth? The means is probability. LDA believes that all thesiss in the world are composed of basic vocabulary. At this time, suppose there is a thesaurus V={v1,v2,...,vn}V={v1,v2,...,vn}, So how to express the theme kk? LDA says to reflect the topic through the probability distribution of words! What a cunning fellow. Let us give an example to illustrate the point of view of LDA. Suppose there is a thesaurus

{Kobe, basketball, football, Obama, Hillary, Clinton}.

Suppose there are two topics

{Sports, politics}.

LDA said that the theme of sports is:

{Kobe: 0.3, basketball: 0.3, football: 0.3, Obama: 0.03, Hillary: 0.03, Clinton: 0.04}

(The number represents the probability of a certain word), and the topic of politics is:

{Kobe: 0.03, basketball: 0.03, football: 0.04, Obama: 0.3, Hillary: 0.3, Clinton: 0.3}

This is how LDA explains what the theme is, and it makes me speechless, and it is very reasonable under careful consideration.

2.3.2 What is the thesis talking about?

Read an thesis for you, and then ask you to briefly summarize what the thesis is talking about. You might answer like this: 80% are talking about politics, the remaining 15% are talking about entertainment, and the rest are nonsense. There are probably three themes that can be extracted here: politics, entertainment, and nonsense. In other words, for a certain thesis, it is very likely that there is not only one topic, but a mixture of several topics. Readers may ask, LDA is a topic model that can be extracted from documents. Did he consider this situation when modeling? Did he forget to consider it? Then you can rest assured, the foresight LDA has already noticed this. LDA believes that there is not necessarily a one-to-one correspondence between thesiss and topics. In other words, thesiss can have multiple topics, and one topic can be in multiple thesiss. I believe that readers can only nod their heads to say yes to this kind of statement. Assuming there are KK topics and MM thesiss, what should be the composition ratio of different topics in each thesis? Since we knew in the last section that we can’t rigidly put a certain word on a certain topic, then of course we can’t talk about a certain topic in a certain thesis, that is, there is such a phenomenon: the same topic, in different thesiss The ratio (probability) of his appearance is different in this. After reading this, readers may have discovered how surprisingly similar the relationship between the document and the topic and the relationship between the topic and the vocabulary are! LDA first stated this discovery. It said that in the previous section we cleverly used the distribution of words to express the topic, so this time is no exception, we cleverly use the distribution of topics to express the thesis! Similarly, let's take an example to illustrate, suppose there are now two thesiss:

"Sports Express", "Entertainment Weekly"

There are three themes

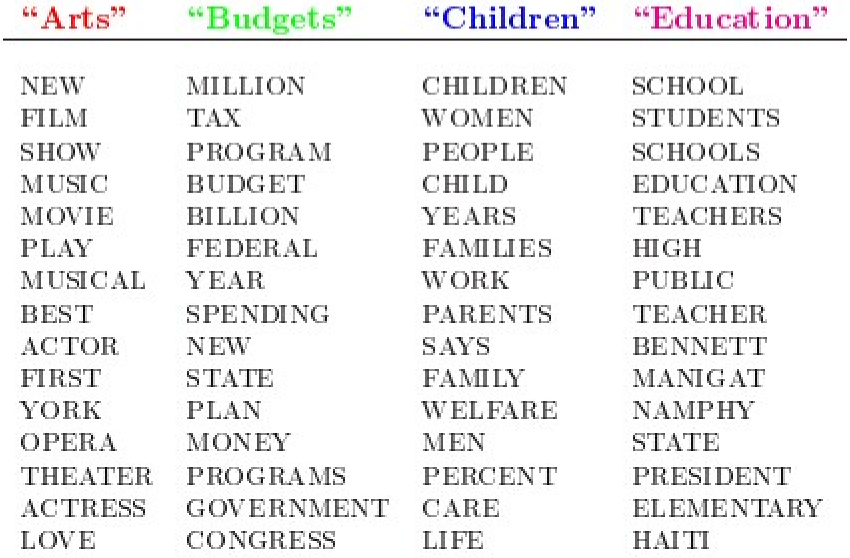
Sports, entertainment, nonsense

So "Sports Express" is like this: [nonsense, sports, sports, sports, sports,..., entertainment, entertainment] and "Entertainment Weekly" is like this: [nonsense, nonsense, entertainment, entertainment, entertainment,..., entertainment, sports]

In other words, what an thesis is talking about can be summarized through different topic ratios.

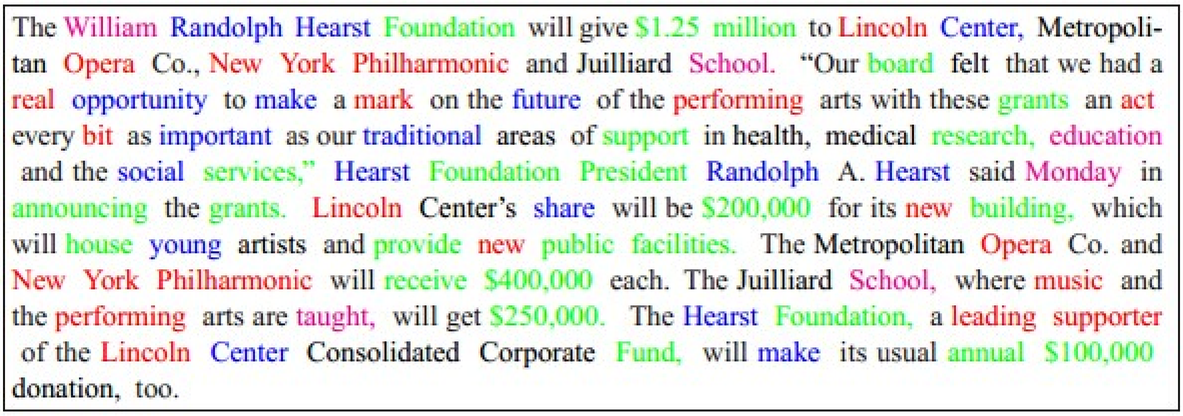
2.3.3 How is the text generated?

How do humans generate documents? The three authors of LDA gave a simple example in the original paper. For example, suppose that these topics are given in advance: Arts, Budgets, Children, Education, and then through learning and training, get the words corresponding to each topic Topic. As shown below:



"Arts"	"Budgets"	"Children"	"Education"
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

Then select one of the above topics with a certain probability, and then select a word under that topic with a certain probability, repeat these two steps, and finally generate an thesis as shown in the figure below (the words in different colors correspond to Words under different themes in the picture above):



The William Randolph Hearst Foundation will give $1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. "Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services," Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center's share will be $200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive $400,000 each. The Juilliard School, where music and the performing arts are taught, will get $250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual $100,000 donation, too.

When we see an thesis, we often like to speculate on how the thesis was generated. We may think that the author first determines the themes of the thesis, and then constructs sentences around these themes and expresses them.

LDA is here to do this: according to a given document, inversely infer its topic distribution.

In layman's terms, it can be assumed that humans have written various thesiss based on the above-mentioned document generation process. Now a small group of people want computers to use LDA to do one thing: your computer will give me speculation and analysis of each thesis on the Internet. What topics have been written, and what is the probability (topic distribution) of each topic in each thesis.

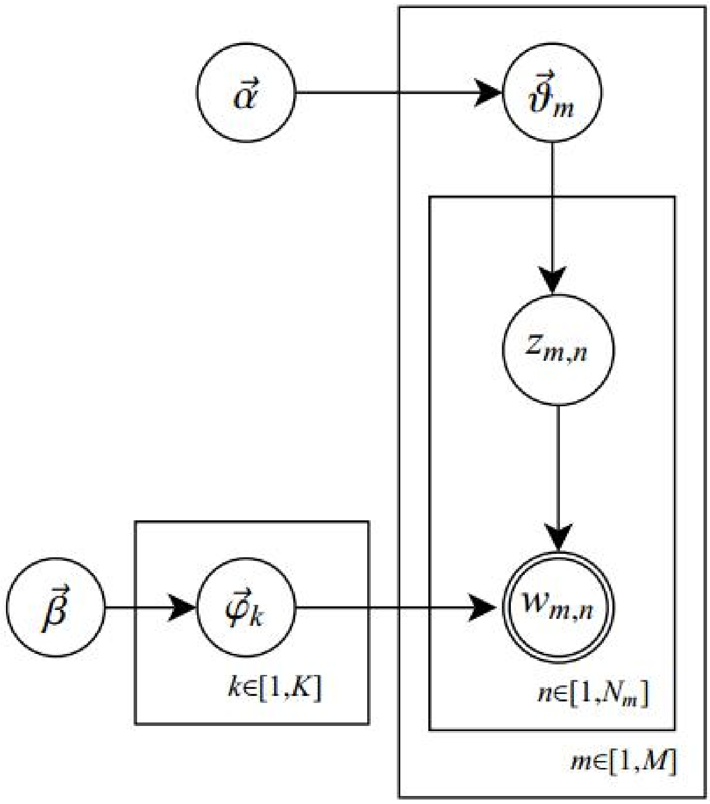
In the LDA model, the way a document is generated is as follows:

a. Sampling from the Dirichlet distribution to generate the topic distribution of document i

b. Sampling from the topic polynomial distribution to generate the topic of the jth word of the document i
c. Sampling from the Dirichlet distribution to generate the word distribution corresponding to the topic
d. Sampling from the polynomial distribution of words and finally generating words

Among them, the similar Beta distribution is the conjugate prior probability distribution of the binomial distribution, and the Dirichlet distribution (Dirichlet distribution) is the conjugate prior probability distribution of the polynomial distribution.

In addition, the graph model structure of LDA is shown in the following figure (similar to the Bayesian network structure):



So if we have a trained LDA model, if we take a new text as input, we can get the topic distribution of the text, we know what kind of thesis it is, and we can also get the distribution of the words under the topic, thereby obtaining the key words of the text.

2.4 Mathematics in LDA

2.4.1 Bag-of-words model

LDA uses a bag-of-words model. The so-called bag-of-words model refers to a document, we only consider whether a word appears, regardless of the order in which it appears. In the bag of words model, "I like you" and "You like me" are equivalent. The opposite of the bag-of-words model is n-gram, which considers the order in which words appear.

2.4.2 Conjugate Prior Distribution

In Bayesian probability theory, if the posterior probability P(θ|x) and the prior probability p(θ) satisfy the same distribution law, then the prior distribution and the posterior distribution are called conjugate distributions. At the same time, the prior distribution is called the conjugate prior distribution of the likelihood function.

The Dirichlet distribution is the conjugate distribution of the polynomial distribution.

Conjugation means, for example, the Dirichlet distribution polynomial distribution. When the data conforms to the polynomial distribution, the prior distribution and posterior distribution of the parameters can maintain the form of the Dirichlet distribution. The advantage of this form is that the prior distribution gives the parameters a clear physical meaning. This physical meaning can be extended to the subsequent distribution for interpretation. At the same time, the knowledge supplemented from the data in the process of transforming from the prior to the posterior can easily have a physical interpretation.

2.4.3 Multinomial distribution

Multinomial distribution is the case where the binomial distribution is extended to multiple dimensions. Multinomial distribution means that the value of the random variable in a single experiment is no longer 0-1, but has a variety of discrete values (1,2, 3...,k). The probability density function is:

$$P(x_1, x_2, \ldots, x_k; n, p_1, p_2, \ldots, p_k) = \frac{n!}{x_1! \ldots x_k!} {p_1}^{x_1} \ldots {p_k}^{x_k}$$

2.4.4 Dirichlet distribution

The probability density function of Dirichlet is:

$$f(x_1, x_2, \ldots, x_k; \alpha_1, \alpha_2, \ldots, \alpha_k) = \frac{1}{B(\alpha)} \prod_{i=1}^{k} {x_i}^{\alpha^i - 1}$$

$$B(\alpha) = \frac{\prod_{i=1}^{k} \Gamma(\alpha^i)}{\Gamma(\sum_{i=1}^{k} \alpha^i)}, \sum_{i=1}^{k} x^i = 1$$

2.4.5 Text modeling

A document can be regarded as a sequence of ordered words. From a statistical point of view, the generation of the document can be regarded as the result of God throwing a dice. Every time a dice is thrown, a vocabulary is generated, and N words are thrown. Generate a document. In statistical text modeling, we hope to guess how God plays this game. This involves two core issues:

1.What kind of dice does God have;

2.How God throws these dice;

The first question is what parameters are in the model, and the probability of each face of the dice corresponds to the parameters in the model; the second question is what the rules of the game are, God may have different types of dice, God These dice can be thrown according to certain rules to generate word sequences.

2.4.6 Unigram Model

In the Unigram Model, we adopt the bag-of-words model, assuming that the documents are independent of each other, and the words in the documents are independent of each other. Assuming that there are a total of V words in our dictionary, the simplest Unigram Model is to think that God produces text according to the following game rules.

1. God has only one dice, this dice has V sides, each side corresponds to a word, and the probability of each side is different;

2. Every time a dice is thrown, the thrown face corresponds to a word; if there are N words in a document, then independently throw the dice n times to produce n words;

2.4.7 Frequent Perspective

For a dice, the probability of each face is denoted as $\vec{p} = (p_1, p_2, \cdots, p_V)$, and each word generated can be regarded as a polynomial distribution, denoted as $\omega \sim Mult(\omega \mid \vec{p})$. A document $d = \vec{\omega} = (\omega_1, \omega_2, \cdots, \omega_n)$, its generation probability is $p(\vec{\omega}) = p(\omega_1, \omega_2, \cdots, \omega_n) = p(\omega_1)p(\omega_2)\cdots p(\omega_n)$.

The documents are considered to be independent. For a corpus, the probability is:

$$W = (\vec{\omega}_1, \vec{\omega}_2, \cdots, \vec{\omega}_m)$$

Assuming that the total word frequency in the corpus is N then,

$\vec{n} = (n_1, n_2, \cdots, n_V)$ obeys the polynomial distribution:

$$p(\vec{n}) = Mult(\vec{n} \mid \vec{p}, N) = \binom{N}{\vec{n}} \prod_{k=1}^{V} p_k^{n_k}$$

The probability of the entire corpus is

$$p(W) = p(\vec{\omega}_1)p(\vec{\omega}_2) \cdots p(\vec{\omega}_m) = \prod_{k=1}^{V} p_k^{n_k}$$

At this point, we need to estimate the parameter p in the model, that is, the probability of each face in the vocabulary dice. According to the point of view of the frequency school, use maximum likelihood estimation to maximize p(W), so the estimated value of parameter pi for

$$\hat{p}_i = \frac{n_i}{N}$$

2.4.8 Bayesian perspective

For the above model, statisticians of the Bayesian School of Statistics will have different opinions. They will be very critical and criticize that it is unreasonable to assume that God has the only fixed dice. In the view of Bayesian school, all parameters are random variables. The dice in the above model is not the only fixed one, it is also a random variable. So according to the Bayesian school of thought, God is playing the game according to the following process:

1. There is a jar with an infinite number of dice, which contains all kinds of dice, and each dice has V faces;

2. Now draw a dice from the jar, and then use this dice to continue tossing until all the words in the corpus are generated

There are an infinite number of dice in the jar, and some types of dice have more dice, and some have fewer dice. From the perspective of probability distribution, the dice in the jar $\vec{p}$ obey a probability distribution $p(\vec{p})$, which is called the prior distribution of parameters $\vec{p}$. From this perspective, we don't know which die is used. Each die may be used, and its probability is determined by the prior distribution. For each specific dice, the probability of generating a corpus from the dice is $p(W \mid \vec{p})$, so the probability of generating a corpus is to sum the corpus generated on each dice.

$$p(W) = \int p(W \mid \vec{p})p(\vec{p})d\vec{p}$$

There are many options for prior probability, but we noticed that

$p(\vec{n}) = Mult(\vec{n} \mid \vec{p}, N)$. We know that polynomial distribution and Dirichlet distribution are conjugate distributions, so a better choice is to use Dirichlet distribution

$$Dir(\vec{p} \mid \vec{\alpha}) = \frac{1}{\Delta(\vec{\alpha})} \prod_{k=1}^{V} p_k^{\alpha_k - 1}, \vec{\alpha} = (\alpha_1, \cdots, \alpha_V)$$

From the polynomial distribution and Dirichlet distribution are conjugate distributions, we can get:

$$p(\vec{p}|W, \vec{\alpha}) = Dir(\vec{p} \mid \vec{n} + \vec{\alpha}) = \frac{1}{\Delta(\vec{n} + \vec{\alpha})} \prod_{k=1}^{V} p_k^{n_k + \alpha_k - 1} d\vec{p}$$

2.4.9 process of pLSA

So how is the document generated in pLSA?

Suppose you want to write M documents. Since a document consists of different words, you need to determine the word in each position in each document.

Suppose you have a total of K optional topics and V optional words. Let's play a game of throwing dice.

1. Suppose you will make a K-sided "document-topic" dice for every document you write (throw this dice to get any of the K topics), and K V-sided "topic-term" dice (Each dice corresponds to a theme, K dice corresponds to the previous K themes, and each side of the dice corresponds to the word item to be selected, and V faces correspond to V optional words).

For example, we can set K=3, that is, make a "document-theme" dice with 3 themes. These 3 themes can be: education, economy, and transportation. Then set V = 3, make 3 "theme-term" dice with 3 sides, where the words on the 3 sides of the education themed dice can be: university, teacher, course, and the 3 sides of the economic themed dice The words can be: market, enterprise, finance, and transportation. The words on the three sides of the dice can be: high-speed rail, car, and airplane.

2. Every time you write a word, first throw the "document-topic" dice to select the topic. After you get the result of the topic, use the "topic-term" dice corresponding to the topic result, and throw the dice to select the word you want to write.

First throw the "document-topic" dice, assuming (with a certain probability) that the topic obtained is education, so the next step is to throw the education topic sieve, (with a certain probability) to get a certain word corresponding to the education topic sieve: university .

The above process of throwing the dice to generate words is simplified as follows: "First select the topic with a certain probability, and then select the word with a certain probability". In fact, there were three themes to choose from at the beginning: education, economy, and transportation. Why did you choose the theme of education? In fact, it is randomly selected, but this random follows a certain probability distribution. For example, the probability of selecting an education theme is 0.5, the probability of selecting an economic theme is 0.3, and the probability of selecting a transportation theme is 0.2. Then the probability distribution of these three themes is {education: 0.5, economy: 0.3, transportation: 0.2}, We call the probability distribution of each topic z appearing in document d the topic distribution, and it is a multinomial distribution.

Similarly, after randomly extracting educational topics from the topic distribution, you are still faced with three words: university, teacher, and curriculum. These three words may be selected, but their probability of being selected is also different. For example, the probability that the word university is selected is 0.5, the probability that the word teacher is selected is 0.3, and the probability that a course is selected is 0.2. Then the probability distribution of these three words is {University: 0.5, Teacher: 0.3, Course: 0.2}, we call the probability distribution of each word w appearing under the topic z as the word distribution, and this word distribution is also a multinomial distribution.

Therefore, the topic selection and word selection are two random processes. First, extract the topic: education from the topic distribution {education: 0.5, economy: 0.3, transportation: 0.2}, and then from the distribution of words corresponding to the education topic {University : 0.5, teacher: 0.3, course: 0.2} Extract the word: university.

3. Finally, you keep throwing "document-topic" dice and "topic-term" dice repeatedly, repeat N times (produce N words), complete a document, repeat the method of generating a document M times , Then complete M documents.

The abstraction of the above process is the document generation model of pLSA. In this process, we did not pay attention to the order of appearance between words and words, so pLSA is a bag-of-words method.

2.4.10 LDA model

In fact, if you understand the pLSA model, you will almost understand the LDA model, because LDA is a Bayesian framework based on pLSA, that is, LDA is the Bayesian version of pLSA (because LDA is Bayesianized Therefore, it is necessary to consider the historical prior knowledge and add the two prior parameters).

Comparison of pLSA and LDA: document generation and parameter estimation

In the pLSA model [7], we follow the following steps to get the "document-term" generative model:

a. Select a document according to probability
b. After selecting the document, determine the topic distribution of the thesis
c. Select an implicit topic category according to probability from the topic distribution

d. After selection, determine the word distribution under the topic
e. Choose a word from the word distribution according to the probability ”

Below, let’s compare the way a document is generated in the LDA model described at the beginning of this thesis [8]:

a. Select a document according to the prior probability
b. Sampling from the Dirichlet distribution (ie Dirichlet distribution) to generate the topic distribution of the document, in other words, the topic distribution is generated by the Dirichlet distribution with the hyperparameter
c. Sampling from the topic polynomial distribution to generate the topic of the jth word of the document
d. Sampling from the Dirichlet distribution (ie Dirichlet distribution) to generate the word distribution corresponding to the topic, in other words, the word distribution is generated by the Dirichlet distribution with the parameter
e. Sampling from the polynomial distribution of words and finally generating words ”

It can be seen from the above two processes that LDA adds two Dirichlet priors to topic distribution and word distribution on the basis of pLSA.

Continue to take the previous example of pLSA to explain in detail. As mentioned earlier, in pLSA, both topic selection and word selection are two random processes. First, extract the topic: education from the topic distribution {education: 0.5, economy: 0.3, transportation: 0.2}, and then from the topic Corresponding word distribution {University: 0.5, Teacher: 0.3, Course: 0.2} Extracted word: University.

In LDA, topic selection and word selection are still two random processes. It is still possible to first extract the topic from the topic distribution {education: 0.5, economy: 0.3, transportation: 0.2}: education, and then from this The word distribution corresponding to the topic {University: 0.5, Teacher: 0.3, Course: 0.2} Extracted word: university.

What is the difference between pLSA and LDA? The difference is:

In pLSA, the topic distribution and word distribution are uniquely determined. It can be clearly pointed out that the topic distribution may be {education: 0.5, economy: 0.3, transportation: 0.2}, and the word distribution may be {university: 0.5, teacher: 0.3, course: 0.2}.

However, in LDA, the topic distribution and word distribution are no longer uniquely determined, that is, they cannot be given exactly. For example, the topic distribution may be {education: 0.5, economy: 0.3, transportation: 0.2}, or it may be {education: 0.6, economy: 0.2, transportation: 0.2}. We don’t know which one it is (that is, we don’t know) because It is random and changeable. But no matter how it changes, it still obeys a certain distribution, that is, the topic distribution and the word distribution are randomly determined by Dirichlet a priori.

2.5 The results

2.5.1 LSA

We ran the LSA algorithm based on all the story data sets. The results are as follows:

Topic 0:

like

know

said

yeah

remember

think

going

people

Topic 1:

like

tell

work

appreciate

people

nerves

tears

childhood

Topic 2:

work

tell

said

people

test

childhood

nerves

passed

Topic 3:

said

laughs

like

says

know

guys

right

crying

Topic 4:

tony

tariq

azim

khamisa

hicks

years

prison

narrator

Topic 5:

like

music

mother

hair

wanted

little

grade

black

Topic 6:

vote

tony

like

school

going

passport

felt

tariq

Topic 7:

mother

miss

school

going

high

love

trying

really

Topic 8:

black

sergeant

military

honor

love

people

major

fallen

Topic 9:

vote

family

passport

face

kind

married

people

tree

In fact, this result is not good, because we cannot determine the number of topics, and the correlation between each group of words is not great. When we have a new data set to join, we must re-run the entire algorithm based on the old data set and the new data set.

2.5.2 LDA

In our experiment, the training set we chose is the standard data set 20news_group, we trained an LDA model based on it, and our test set is some stories downloaded from https://storycorps.org/. These data are all unlabeled. We have selected some results to display here. Through the results, we can determine what kind of text it belongs to and what the keywords of this text are.

The results of our LDA training based on the training set are as follows:

```
[(0,
  '0.034*"drive" + 0.019*"disk" + 0.013*"use" + 0.010*"scsi" +
0.010*"system" + 0.009*"tape" + 0.008*"controller" + 0.007*"card"'),
 (1,
  '0.025*"space" + 0.010*"launch" + 0.008*"satellite" + 0.008*"nasa" +
0.006*"ripem" + 0.006*"mission" + 0.005*"orbit" + 0.005*"earth"'),
 (2,
  '0.032*"game" + 0.024*"team" + 0.020*"play" + 0.016*"player" +
0.014*"win" + 0.011*"season" + 0.011*"year" + 0.009*"league"'),
 (3,
  '0.026*"use" + 0.013*"windows" + 0.013*"card" + 0.011*"problem" +
0.011*"run" + 0.011*"work" + 0.011*"know" + 0.010*"system"'),
 (4,
  '0.005*"scsi-1" + 0.004*"sharks" + 0.003*"scsi-2" + 0.003*"fleet" +
0.003*"goaltender" + 0.002*"torque" + 0.002*"snap" + 0.002*"isaiah"'),
 (5,
  '0.027*"gun" + 0.015*"law" + 0.014*"state" + 0.013*"right" +
0.011*"government" + 0.010*"weapon" + 0.009*"crime" + 0.008*"use"'),
 (6,
  '0.030*"file" + 0.016*"use" + 0.014*"program" + 0.012*"entry" +
0.011*"window" + 0.009*"image" + 0.008*"server" + 0.007*"include"'),
 (7,
  '0.014*"health" + 0.010*"drug" + 0.009*"disease" + 0.009*"use" +
0.007*"april" + 0.007*"report" + 0.007*"tobacco" + 0.006*"center"'),
 (8,
  '0.012*"mov" + 0.009*"copy" + 0.008*"rider" + 0.007*"cover" +
0.007*"cubs" + 0.006*"dod" + 0.006*"appear" + 0.005*"man"'),
 (9,
  '0.006*"class" + 0.005*"system" + 0.005*"base" + 0.005*"morris" +
0.005*"think" + 0.004*"lot" + 0.003*"group" + 0.003*"xlib"'),
 (10,
  '0.007*"dos" + 0.007*"adl" + 0.006*"station" + 0.005*"option" +
0.005*"cluster" + 0.004*"area" + 0.004*"arab" + 0.004*"cost"'),
 (11,
  '0.012*"people" + 0.010*"armenians" + 0.009*"israel" + 0.008*"jews" +
0.008*"war" + 0.007*"kill" + 0.006*"come" + 0.005*"woman"'),
 (12,
  '0.010*"picture" + 0.007*"coli" + 0.005*"candida" + 0.004*"captain" +
0.004*"onur" + 0.004*"yalcin" + 0.004*"bob" + 0.004*"sea"'),
 (13,
  '0.010*"mail" + 0.010*"post" + 0.010*"list" + 0.009*"use" +
0.009*"information" + 0.008*"send" + 0.008*"book" + 0.007*"program"'),

 (14,

  '0.012*"year" + 0.012*"think" + 0.011*"know" + 0.009*"time" +
0.009*"people" + 0.007*"work" + 0.006*"day" + 0.006*"come"'),

 (15,

  '0.020*"use" + 0.009*"ground" + 0.008*"power" + 0.007*"wire" +
0.007*"line" + 0.007*"keyboard" + 0.007*"unit" + 0.006*"bit"'),

 (16,

  '0.015*"god" + 0.015*"people" + 0.012*"think" + 0.011*"know" +
0.009*"believe" + 0.008*"thing" + 0.007*"jesus" + 0.007*"mean"'),

 (17,

  '0.057*"g)r" + 0.046*"g9v" + 0.015*"giz" + 0.005*"fij" + 0.005*"c8v" +
0.003*"pitcher" +
0.002*"m"`@("`@("`@("`@("`@("`@("`@("`@("`@("`@("`@("`@("`@("`
@("`@" + 0.002*"m9v"'),

 (18,
  '0.021*"car" + 0.011*"use" + 0.011*"buy" + 0.011*"look" + 0.010*"know"
+ 0.008*"problem" + 0.008*"sell" + 0.008*"think"'),
 (19,
  '0.027*"key" + 0.018*"use" + 0.013*"chip" + 0.013*"encryption" +
0.012*"government" + 0.010*"system" + 0.009*"security" +
0.008*"clipper"')]
```

In fact, pay attention to item 17, a series of meaningless words appeared in our results. We originally thought it was a mistake, but later we discovered that there are some meaningless words in the original data set. They are the noise of the data set, so they are naturally classified into the same category.

Our input text is:

alexia dukes (ad): it feels like you’re running into a burning building and you hope to god that you’re going to come out on the other side. it’s not a feeling i thought i was going to have teaching.

I had a student and he’s like, “i’m so sorry. normally, i’m a really good student. i just, i can’t. i’m really trying.” and so i said, “ok, let me tell you the truth. i just can’t either. i want you to know that what you’re doing isn’t wrong, it’s human.”

maria rivera (mr): they just want to hear, like, everything’s going to be okay. we’re here together and they need that in human being form. i miss those human interactions a lot.

ad: i remember times where i was, like, bawling tears and something i’ve always admired about you as a teacher, and just as a person, is you have this unique ability to — even in the darkest times — just tell people it’s going to be okay.

mr: your biggest gift to me is just being supportive and reminding me that i am this person because it’s been unclear sometimes and i don’t think without your energy i could have made it teaching through this whole pandemic.

i’m extremely thankful that we shared this time together and i just appreciate you so much.

Our output is：

```
[(1, 0.12482031),

 (3, 0.01902153),

 (6, 0.01276091),

 (10, 0.011931259),

 (11, 0.050757196),

 (12, 0.038533688),

 (14, 0.40786138),

 (16, 0.15170461),

 (18, 0.17774184)]
```

This means that this text is composed of 0.12*topic1+0.41*topic14+0.15*topic16+0.18*topic18, which is basically in line with the original text.

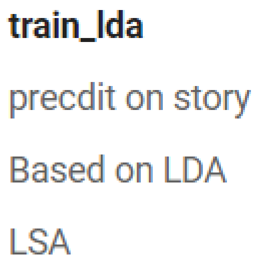
We extracted the first eight words with the highest weights under a topic as the keywords of the text, namely:

“space" ,"launch" ,"satellite" ,"nasa",orbit" ,"earth” ,"year" ,"think" ,"know" ,"time" ,"people" ,"work" ,"day" ,"come"'

2.6 How to use our software?

First, you need to get our story data set. Our story data set is provided by Dennis, and I would like to express my gratitude to Dennis. I uploaded the original version of 120 stories to this link. You need to download this compressed package, unzip it and upload it to the My Drive/temp/AllStories location of your Google Drive. Then run to open the colab file under this link and run it. Next, I will introduce you to the role of each part:

The main body of the program is divided into four parts:



train_lda
precdit on story
Based on LDA
LSA

Among them, train_lda is a process of training an LDA based on 20news_group, and predict on story is a process of prediction based on the test set. Based on LDA is where we directly use our story data set as part of the training set. LSA is our part of running LSA based on the story data set.

I have written the notes of the program in the key parts, and you can watch the notes to get the final result. You can also try to change some of these hyperparameters to get better results.

2.7 Summary

In general, this paper uses the traditional LDA method based on statistics. This method runs fast and can process large quantities of text in a short time. These methods basically solve the current problem, but our method of processing text is relatively simple and when we have more story data, this algorithm may face some other challenges. The biggest challenge is when some of the topics in our story are not available. In the training focus, our results may be inaccurate. For the first problem, future students can solve it by learning more and more complex text processing methods and collecting more data. For the second problem, in fact, I am doing this paper while doing another similar project with my friend. This project is based on "Improving Neural Topic Models using Knowledge Distillation" [9] in this thesis, using the method of knowledge distillation [10] to transfer the knowledge of the teacher model (Bert [11]) to the student model (Scholar), by reducing the value of the student model Deviations to improve its performance. Its brief introduction is as follows:

The purpose of this project is to teach the learned knowledge of the teacher model to the student model through knowledge distillation, thereby reducing model deviation and improving generalization ability. This project is based on the English data set, using BERT as the teacher model, text as input, word counts as the label, outputting the BoW predicted by BERT, and inputting this as knowledge to the student model. The final loss function is the sum of soft loss and hard loss. The soft loss is the difference between the predicted value of the student model and the BoW from BERT, and the hard loss is the difference between the predicted value of the student model and the BoW of the actual text. In comparison, the student model with knowledge distillation doubled the npmi of the top ten topics predicted by the original student model, from 0.17 to 0.34.

At the same time, I also attached a brief introduction of this project at the end of the paper. Hope it will help in future work.The link of the project is this.

3 The further work

3.1 Introduction

In reality, there are three main factors that really restrict the performance of the model, namely data, algorithms and computing power.

It is generally believed that the data represents the upper limit of the model, and our algorithm cannot learn what is not contained in the data. Especially in NLP tasks, such as topic model tasks, we cannot collect a data set that contains all the topics. It is too expensive and unlabeled. Just like our task, when some topics are not included in our training set, our test set will not get a better result anyway. So here we adopt the method of knowledge distillation to teach our topic model the knowledge of a pre-trained model.

Pre-training model Pre-training is to first train the model on a large number of general corpus, learn general language knowledge, and then perform transfer training for the task in a targeted manner. The commonly used word2vec or GloVe word vectors in the past, as well as the current BERT family, are all typical pre-trained language models.

The pre-training method is to use a huge amount of non-annotated corpus design tasks for self-supervised learning-such as w2v's CBOW and BERT's MLM. It should be noted here that self-supervised learning is also a type of supervised learning, but the label comes from the data itself.

From a mathematical point of view, the machine learning algorithms we use daily (including LDA) are a complex mapping from data space to solution space. The LDA we used earlier is a relatively simple mapping.

Later, due to the development of neural networks, people invented new topic models based on neural networks. People also invented Transformer based on the attention mechanism, which is also the basis of the Bert family of the most successful NLP pre-training models so far.

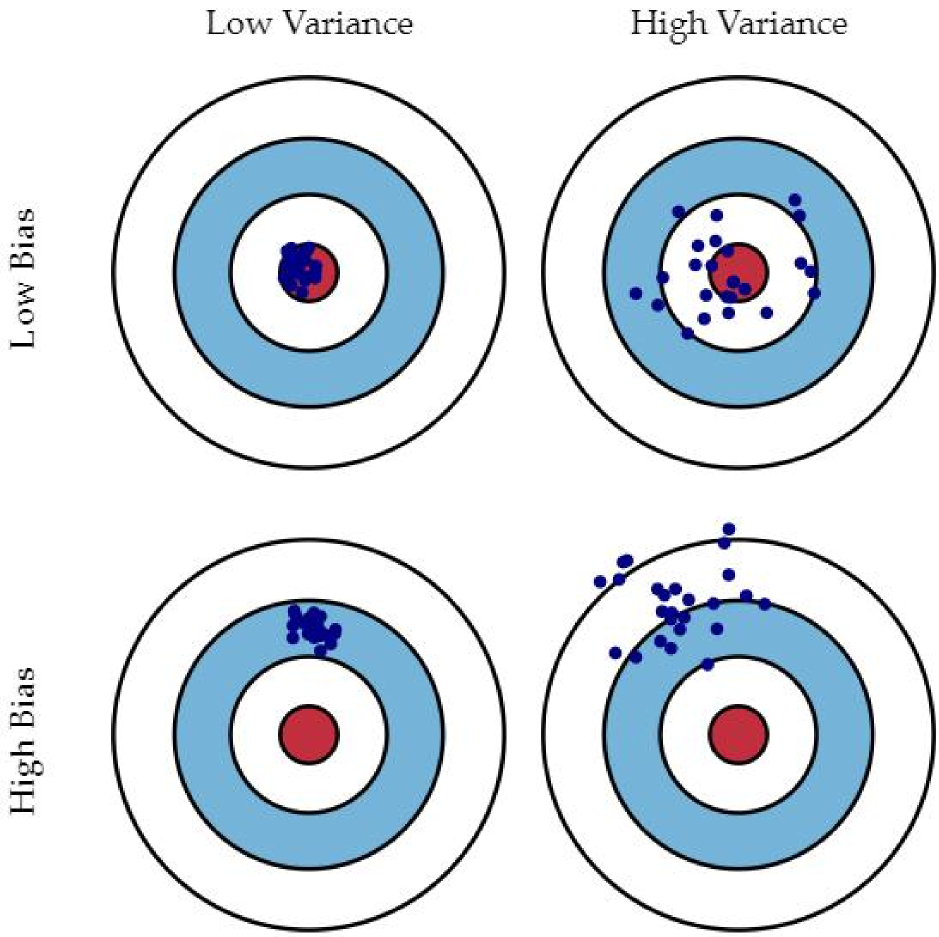
When we train the model, we use the GPU provided by Google for free. Because of the huge amount of data, we spent six hours fine-tuning our Bert. If we want better results, we still need longer training.

3.2 Why does our method work?

Why does our method work? First, let's first introduce what deviation is. If we can obtain all possible data sets and minimize the loss on this data set, then the learned model can be called the "real model". Of course, in real life, it is impossible for us to obtain and train all possible data, so the "real model" certainly exists, but it cannot be obtained. Our ultimate goal is to learn a model to make it closer to the real model.

Bias and Variance describe the gap between the model we learned and the real model from two aspects.

Bias is the difference between the average value of the output of all models trained with all possible training data sets and the output value of the real model.



The bias of our own training model is relatively large. Because the pre-training model is pre-trained on a large-scale data set, its bias is relatively small. If we use the method of knowledge distillation to combine these two models, then the deviation of our topic model will become smaller, thereby improving the performance of the model.

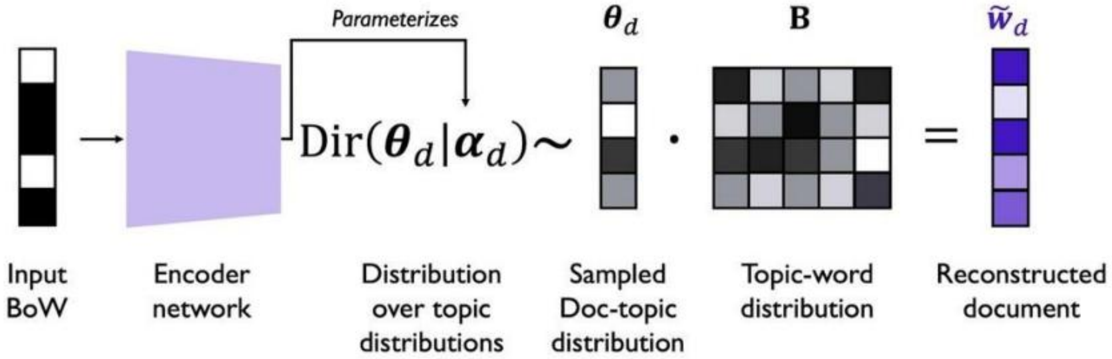
Our specific approach is as follows:

Use BERT as the teacher model, text as input, word counts as label, output the BoW predicted by BERT, and input this as knowledge to the student model. The final loss function is the sum of soft loss and hard loss. The soft loss is the difference between the predicted value of the student model and the BoW from BERT, and the hard loss is the difference between the predicted value of the student model and the BoW of the actual text. Finally, we try to reduce this loss.

3.3 The student model

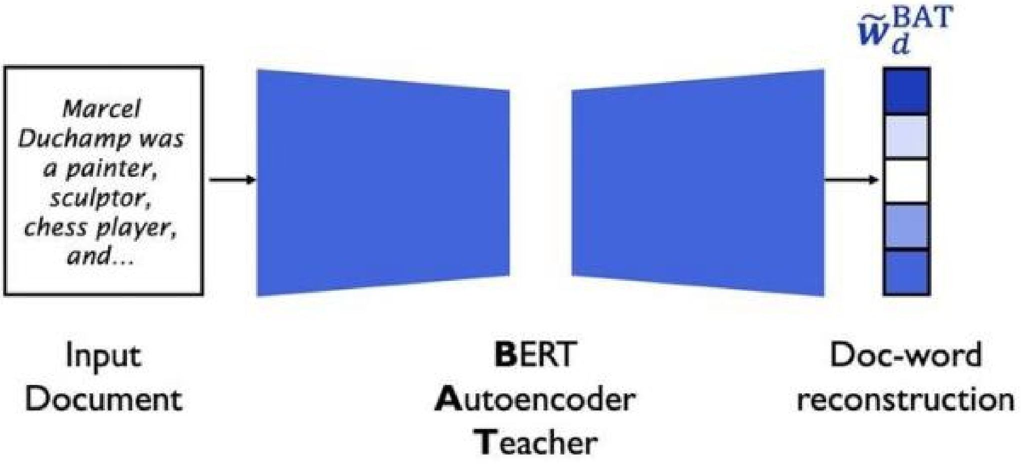
The student model performed a model reconstruction process.

The idea of document reconstruction is similar to that of autoencoder.Encoder produces a Dirichlet distribution over topics, instead of Gaussian in VAE (Variational Autoencoder).Decoder part is exactly the document reconstruction task, as mentioned in previous slides.we casted topic modeling from an unsupervised task to multi-label classification .

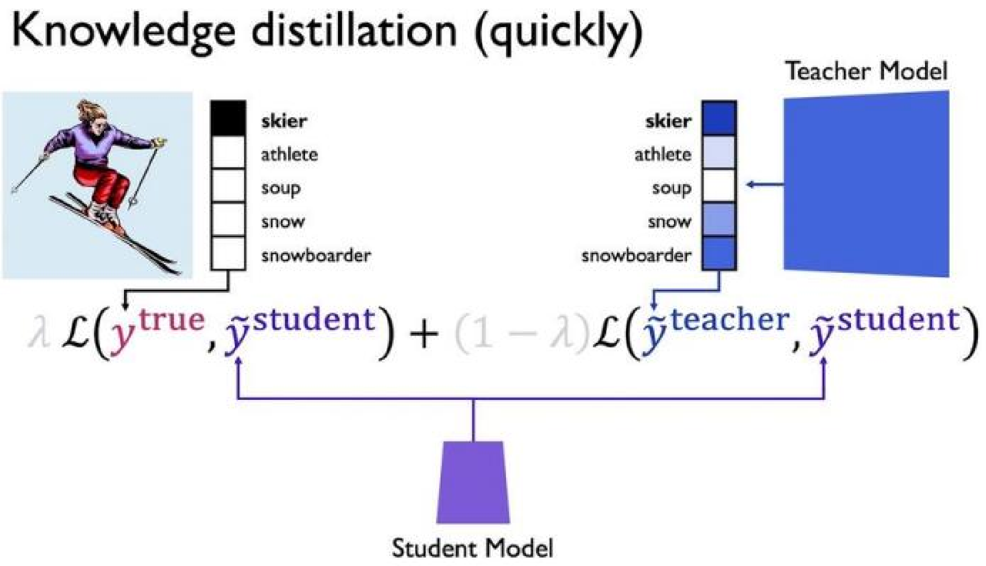


3.4 The teacher model

BERT is a huge pretrained language model.Thanks to its great performance and strong versatility, we can also make BERT a neural topic model after fine-tuning.


$\tilde{w}_d^{\mathrm{BAT}}$
Marcel Duchamp was a painter, sculptor, chess player, and...
Input Document
BERT Autoencoder Teacher
Doc-word reconstruction

There is a basic assumption:Some of the knowledge learnt by a model exists in the logit layer, which usually appears as the estimated probability of each label.Accepting this assumption, it is possible to compress knowledge from a huge teacher model to a simpler student.



Knowledge distillation (quickly)

$\lambda\,\mathcal{L}(y^{\text{true}}, \tilde{y}^{\text{student}}) + (1-\lambda)\mathcal{L}(\tilde{y}^{\text{teacher}}, \tilde{y}^{\text{student}})$

This is also the whole process of knowledge distillation.

$$\begin{aligned}
\boldsymbol{w}_d^{\mathrm{BAT}} &= \sigma\left(\boldsymbol{z}_d^{\mathrm{BAT}}/T\right) N_d \\
\hat{\boldsymbol{w}} &= f\left(\boldsymbol{\theta}_d, \mathbf{B}; T\right) \\
\mathcal{L}_{KD} &= \lambda T^2 \left(\boldsymbol{w}_d^{\mathrm{BAT}}\right)^{\top} \log \hat{\boldsymbol{w}} + (1-\lambda)\mathcal{L}_R
\end{aligned}$$

3.5 Entire process

The entire process of our program is as follows:

a. Fetch 20 newsgroup dataset from sklearn

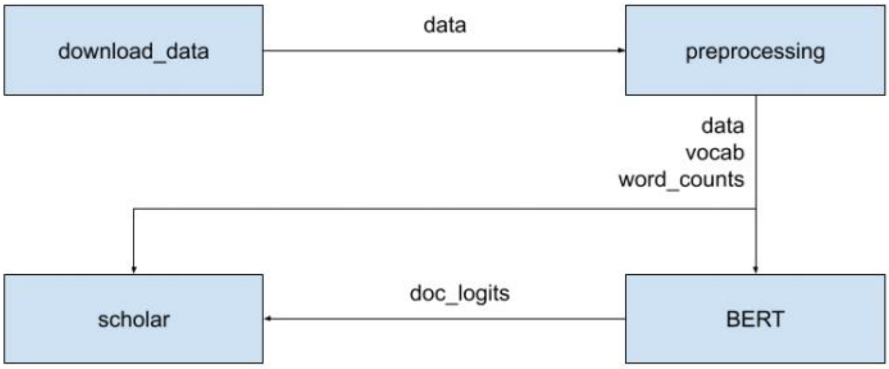
b. After preprocess, pass data to BERT and fine-tune on document reconstruction task

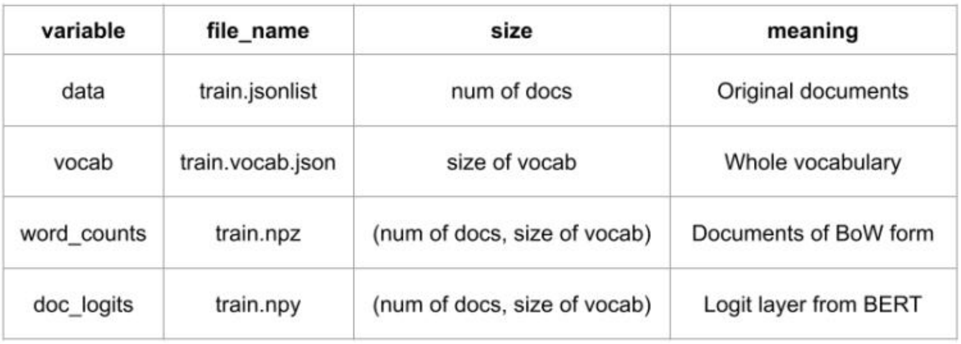
c. Collect the logits from BERT

d. Run Scholar with knowledge distillation

e. Compute NPMI from Palmetto and evaluate the results

The flow chart is as follows:



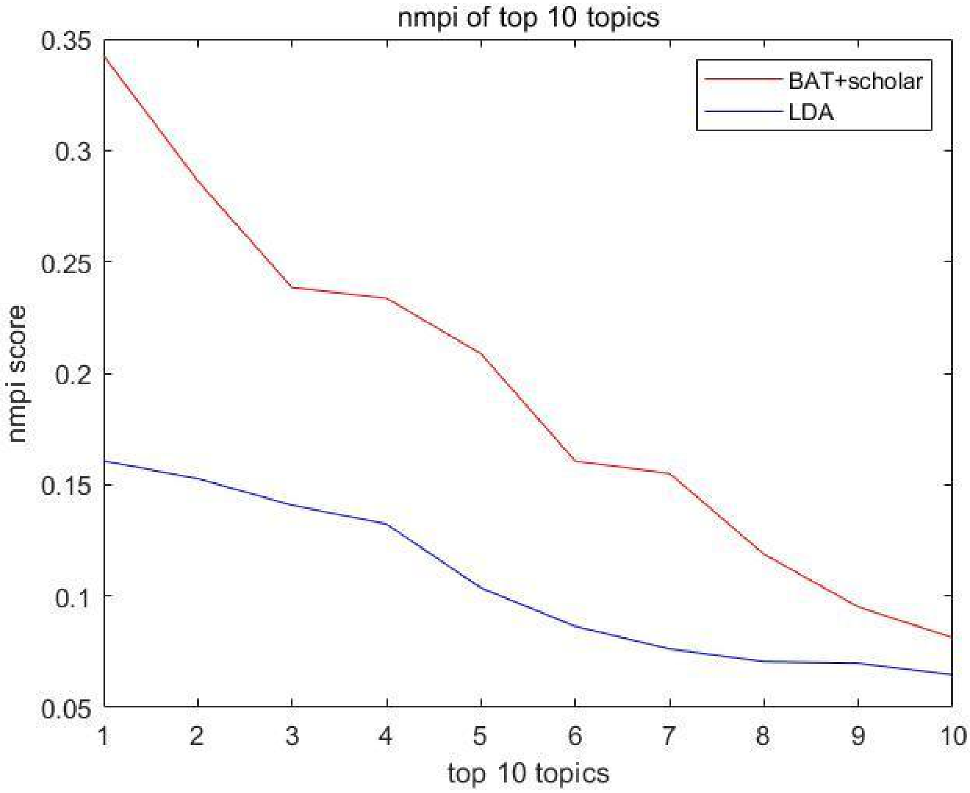


variable	file_name	size	meaning
data	train.jsonlist	num of docs	Original documents
vocab	train.vocab.json	size of vocab	Whole vocabulary
word_counts	train.npz	(num of docs, size of vocab)	Documents of BoW form
doc_logits	train.npy	(num of docs, size of vocab)	Logit layer from BERT

3.6 The results

We finally compared the results of this model with the results of LDA, where the metric is NPMI.

We compared the npmi scores of the top ten topics of BAT+scholar and LDA.BAT+scholar has better performance than LDA.



We picked the two groups as a detailed comparison.

	BAT+scolar	LDA
1	jesus christ doctrine god scripture faith bible salvation	jesus bible christ christian word christians death john
2	espn playoff penguins playoffs rangers detroit leafs devils	game players hockey play team player season points

	BAT+scolar	LDA
1	0.286	0.14
2	0.095	0.016

From the above results, BAT+scholar can get better results than LDA, reflecting the powerful performance of the neural topic model. As a pre-training model, BERT has strong generalization capabilities and can guide the scholar model to give better results. Although the performance of LDA is slightly insufficient, it runs fast, occupies less RAM, and the mathematical theory is relatively complete. It also occupies a place in topic modelling.

3.7 Summary

a. Topic modeling, an unsupervised learning task, can be transformed into document reconstruction, a self-supervised task, which is much like a multi-label classification one.
b. The core of a topic model is $\mathbf{B}$ and θd.
c. Knowledge distillation can be used to transfer knowledge from teacher model to student model
d. Poor-performing teacher model can harm the performance of student model

4.Conclusion

In our thesis, we use topic models to calculate the topic distribution and word distribution of a model. We have tried LSA, LDA based on statistical methods and topic models based on knowledge distillation. Among them, LSA performs better in the face of small batches of data, but SVD requires a longer time, and when new text appears, it is necessary to rerun the algorithm based on the entire data set. LDA is a generative algorithm based on Bayesian assumptions, and the trained model runs fast. But when the test set and the training set do not overlap, the running result is poor. Compared with LDA, the topic model based on knowledge distillation can learn the general laws of the language contained in the pre-training model, so it performs better. But it requires greater running time and computing power.

Acknowledgements

In this part, I would like to thank my professors Luciano Lavagno and Denis Patti. With their help, I completed this thesis. In the context of Corona, we are located in three different countries and have completed an international cooperation. I hope we will have the opportunity to cooperate in the future.I hope this paper can also help this project and help more people record their own stories.

Reference

1. Papadimitriou, C hristos; Raghavan, Prabhakar; Tamaki, Hisao; Vempala, Santosh (1998). Latent Semantic Indexing: A probabilistic analysis. Proceedings of ACM PODS.
2. Hofmann, T. (1999). Probabilistic Latent Semantic Analysis. Uncertainity in Artificial Intelligence.

3. Bellegarda, J. R. (2000). Exploiting latent semantic information in statistical language modeling. Proceedings of the IEEE, 88(8), 1279-1296.
4. Hofmann, T. (2001). Unsupervised Learning by Probabilistic Latent Semantic Analysis. Machine Learning. 42(1-2): 177-196.
5. Blei, David M.; Andrew Y. Ng; Michael I. Jordan (2003). "Latent Dirichlet Allocation" (PDF). *Journal of Machine Learning Research*. 3: 993–1022. doi:10.1162/jmlr.2003.3.4-5.993
6. Distilling the knowledge in a neural network G Hinton, O Vinyals, J Dean - arXiv preprint arXiv:1503.02531, 2015 - arxiv.org
7. Improving neural topic models using knowledge distillation A Hoyle, P Goel, P Resnik - arXiv preprint arXiv:2010.02377, 2020 - arxiv.org
8. Devlin, Jacob; Chang, Ming-Wei; Lee, Kenton; Toutanova, Kristina. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2018-10-11. arXiv:1810.04805v2